**Notification for EOHS Input Inspection**

Taki Guan

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Changes |
| 1.0 | Taki Guan | 2019-05-22 | Create Document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[Version History 1](#_Toc4068331)

[Purpose 1](#_Toc4068332)

[Procedure 1](#_Toc4068333)

[Architecture 2](#_Toc4068334)

[Reference Link 5](#_Toc4068335)

# Purpose

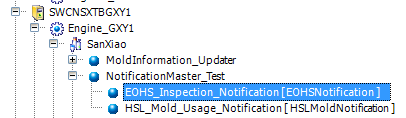
This is a requirement from EOHS team if operator input some specific item in the failed input field. System will check if there are some key items inside. If it’s true, system will automatically generate an email and send to some specific person.

# Procedure

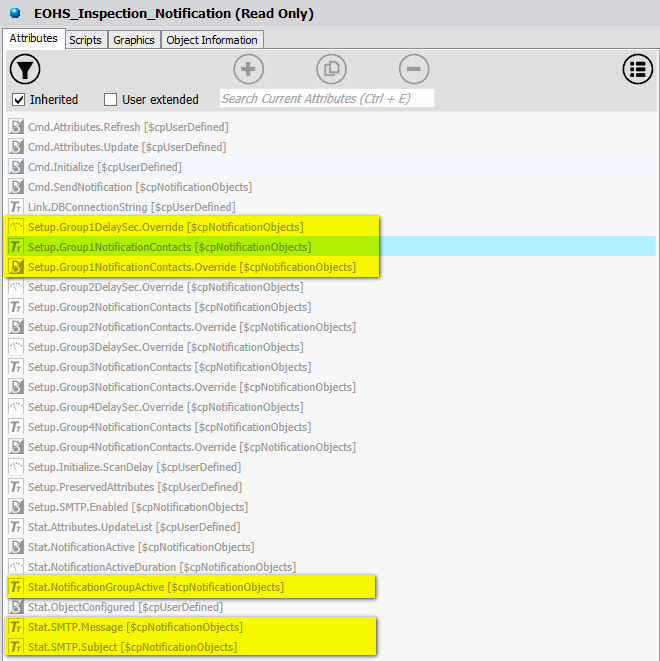
EOHS inspection item 1 and 2 are the key monitor items. The business requirement is when user input value include 1 or 2. We detect what they input and then send email notifications.

Here we want to use regex expression to match all numbers user input. The loop the result set to find if number 1 or 2 are inside or not.

We use a message object to handle email subject, contents, contacts and delay seconds.



[EOHS\_Inspection\_Notification] will be deployed under galaxy server to center control all EOHS inspection notifications.



These attribute need to be specified for the requirement.

* [Setup.Group1DelaySec.Override] need to be set to [0] if notification need to be sent immediately.
* [Setup.Group1NotificationContacts.Override] need to be set to [True] if we want to specify contacts for this specific object.
* [Setup.Group1NotificationContacts] need to maintain if you don’t want to use contacts from Notification Master object.
* [Stat.NotificationGroupActive], [Stat.SMTP.Message] and [Stat.SMTP.Subject] need to be specified in graphic script

# Architecture

Architecture and flow are as below flowchart. Code is designed base on the flowchart.



First, we need to create a control attribute [b\_SendEOHSMessage] to make sure operator has finished input.

If [b\_SendEOHSMessage] has been set to true, script [Script\_Send\_EOHS\_Message] will work as below code shows.

*'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''*

*' Script Name:* *Script\_Send\_EOHS\_Message*

*' Expression: b\_SendEOHSMessage*

*' Trigger: OnTrue*

*'===================================================================================*

*' Author: Taki Guan <taki.guan@colpal.com>*

*' Creation Date: 2019-5-21*

*' Description: Send EOHS Notifications base on input contents*

*' Pre-condition:*

*' Post-condition:*

*'-----------------------------------------------------------------------------------*

*' Revision History:*

*' 000 - 2019-05-21 - Taki Guan <taki.guan@colpal.com>*

*' Initial Creation*

*' 001 - 2019-05-22 - Taki Guan <taki.guan@colpal.com>*

*' Add comments for scripts*

*''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''*

b\_SendEOHSMessage = false;

*'Variable definition*

dim pattern as string;

dim matches as System.Text.RegularExpressions.MatchCollection;

dim match as System.Text.RegularExpressions.Match;

dim messageSent as System.Text.StringBuilder;

*'Assign value to variables*

pattern = "\d+";

messageSent = new System.Text.StringBuilder();

try

    if s\_CsxDL10Value2 <> null then

*'LogMessage("Before Split " + s\_CsxDL10Value2);*

*'Return any numbers of input abnormal EOHS value*

        matches = System.Text.RegularExpressions.Regex.Matches(s\_CsxDL10Value2, pattern);

        for each match in matches

            try

                if System.Int32.Parse(match.Value) == 1 then

                    if messageSent.Length == 0 then

                        messageSent.AppendLine("01 固定防护罩 安全检查失效");

                    endif;

*'LogMessage(messageSent.ToString());*

                endif;

                if System.Text.RegularExpressions.Regex.IsMatch(match.Value, "2") then

                    messageSent.AppendLine("02 安全门联锁 安全检查失效");

                    messageSent.AppendLine("此邮件由OPERA系统自动发出，请自行验证正确性。");

*'LogMessage(messageSent.ToString());*

                endif;

            catch

                LogError("Script\_Send\_EOHS\_Message Match Failed");

            endtry;

        next;

        if messageSent.Length <> 0 then

*'Call method from Notification object*

            EOHS\_Inspection\_Notification.Stat.SMTP.Subject = System.String.Format("{0} 机台安全检查", InTouch:MES\_Cur\_Line);

            EOHS\_Inspection\_Notification.Stat.SMTP.Message = messageSent.ToString();

            EOHS\_Inspection\_Notification.Stat.NotificationGroupActive = "Group1";

            EOHS\_Inspection\_Notification.Stat.NotificationActive = true;

        endif;

*'LogMessage(" Script\_Send\_EOHS\_Message Split Successfuly " + matches);*

    endif;

catch

    LogError("Script\_Send\_EOHS\_Message Split Failed");

endtry;

Here we user regex pattern [\d+] to match all number input. Then we use .NET framework method [System.Text.RegularExpressions.Regex] to match user input value. The return result is a match collection, we need to use data type [MatchCollection] to store data.

# Reference Link

1. [.NET Regular Expressions](https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular-expressions)
2. [Regular Expression Language - Quick Reference](https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference)